

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

**Methods and Systems for Context-Aware Policy
Determination and Enforcement**

Inventor(s):

Gopal Parupudi

Stephen S. Evans

Edward F. Reus

ATTORNEY'S DOCKET NO. MS1-697

1 **TECHNICAL FIELD**

2 This invention relates to generally to the area of context-aware computing
3 or ubiquitous computing.
4

5 **BACKGROUND**

6 The World Wide Web (WWW) was created to make content available from
7 any source in any location around the world. Users of the Web are able to
8 generally access a seemingly infinite number of resources via the Web. The Web
9 has been highly successful in this regard. Yet, with the evolution of the Web,
10 certain needs remain largely unmet. Specifically, people continue to have a need
11 to access information that has a contextual aspect to it. That is, often times,
12 individuals will find themselves in a computing environment that carries with it a
13 certain context. Yet, the context of the environment cannot be easily incorporated
14 into the present computing environment. As an example, consider the context of
15 location. People generally have a need to access information, data, resources and
16 the like, that have geographic dimensions to them. For example, individuals may
17 desire to take advantage of services or products that are close in proximity to
18 where they currently are located. In this regard, it is desirable to understand the
19 individual's contextual location so that services, goods and the like can be made
20 available to the individual. As "eCommerce" continues to grow in importance, the
21 necessity of bringing people, places, services and goods together in an efficient
22 manner will become critically important.

23 To date, many attempts have been made to bring people, places, services
24 and goods together. These various attempts have generally approached the
25 problem from different directions in an often times incompatible manner. As an

example, consider the context of location. Some services have attempted to bring people and services together by defining large databases that maintain information about the services. For example, a list of restaurants may be maintained in a web accessible database where each restaurant is associated with a zip code in which the restaurant is located. When a user desires to locate a particular restaurant, they might simply enter the zip code where they are located to see a list of corresponding restaurants in that zip code. From the list of restaurants, they might be able to select one or two restaurants of interest. This approach is undesirable for a number of reasons. First, the operation of the system is dependent upon a central server that is responsible for receiving user queries and executing the queries to return the information to the user. In the event the server fails, so too does the service. In addition, this particular service might be suited to finding restaurants, but possibly not other businesses. In addition, the granularity with which the results are returned to the user may foist some of the search burden on the user (i.e. the user gets a list of restaurants in a nearby zip code, but has to further explore the list to select which ones are of interest). Further, the list of restaurants may include some restaurants that are blocked by some type of a physical barrier (i.e. a river, mountain, etc.) that makes the distance, as the crow flies, unroutable.

Providers of services and products want to be connected to nearby end-users. End-users want to consume these services and goods at the closest and most convenient location. Acquiring the services of a dentist or a plumber that lives somewhere "out on the net" is not appropriate if you need them to fill a cavity or unclog a sink. Looking for the nearest hotdog while in a stadium requires you to stay in the stadium.

There is an unsolved need to be able to create context-aware computing in which computing devices can participate in their particular context. In specific circumstances, there are needs to provide relational position awareness among physical locations in both public and private views of the world. To date, however, there is no one standardized view of the world that would unlock the potential of context-aware computing. Context-aware computing is much more than just position awareness—although this is a very big field in and of itself.

This invention arose out of concerns associated with developing a standardized, context-aware infrastructure and related systems to unlock the potential of context-aware computing.

SUMMARY

Context aware computing systems and methods are described. In one described embodiment, devices and methods are provided that are context-aware (in one example—location-aware) in that they provide for the application and enforcement of various policies as a function of context. Specifically, various computing devices, through the described methodologies and structures, are able to automatically determine their context. Once context is determined, a collection of policies can be evaluated to provide a resultant set of policies that apply to the given context. The resultant set of policies are then enforced, typically via the device's operating system. Policy enforcement can involve promulgating new settings or state to applications that are executing on or off the device. Advantageously, the devices and methodologies can adapt the resultant set of policies as the device's context changes so that the policies can be dynamically determined and enforced automatically as the device's context changes.

1
2 **BRIEF DESCRIPTION OF THE DRAWINGS**

3 Fig. 1 is a diagram of an exemplary computing device that can be used in
4 accordance with the described embodiments.

5 Fig. 2 is a conceptual diagram of an exemplary Master World and an
6 exemplary Secondary World in accordance with the described embodiment.

7 Fig. 3 is an exemplary specific view of a Master World and a Secondary
8 World and their relation to one another.

9 Fig. 4 is a flow diagram that describes steps in a method in accordance with
10 the described embodiment.

11 Fig. 5 is a flow diagram that describes steps in a method in accordance with
12 the described embodiment.

13 Fig. 6 is a high level diagram of an exemplary computing device
14 architecture.

15 Fig. 7 is a somewhat more specific view of an exemplary computing device
16 architecture.

17 Fig. 8 is a flow diagram that describes steps in a method in accordance with
18 the described embodiment.

19 Fig. 9 is a flow diagram that describes steps in a method in accordance with
20 the described embodiment.

21 Fig. 10 is a flow diagram that describes steps in a method in accordance
22 with the described embodiment.

23 Fig. 11 is a side elevational view of an exemplary location beacon in
24 accordance with one embodiment.
25

1 Fig. 12 is a block diagram that illustrates an architecture in accordance with
2 one described embodiment.

3 Fig. 13 is a block diagram that illustrates a policy collection that can be
4 provided in accordance with one described embodiment.

5 Fig. 14 is a flow diagram that describes steps in a method in accordance
6 with the described embodiment.

7 Fig. 15 is a flow diagram that describes steps in a method in accordance
8 with the described embodiment.

9 10 **DETAILED DESCRIPTION**

11 **Overview**

12 To provide a standardized solution, embodiments described just below
13 provide a uniform definition of the world. The uniform definition is defined in
14 terms of a hierarchical tree of nodes, where each node represents some aspect of
15 the world. Each node is connected to at least one other node by a branch. An
16 exemplary classification of nodes takes place on a physical level (e.g. physical
17 locations such as political entities, infrastructure entities and public places), as
18 well as a non-physical level (e.g. military APOs). This hierarchical nodal structure
19 is referred to as the Master World, and is a standardized view worldwide. Each
20 node of the Master World has various attributes associated with it that assist in
21 context-aware computing. Exemplary attributes include a unique ID, name,
22 geographic entity class, latitude/longitude, relative importance, contextual parents
23 to name just a few. The Master World is useful because it can be used to
24 determine the relative location of a place anywhere in the world and at any
25 definable granularity.

1 Once an individual's location or a place an individual is interested in is
2 determined, various services that reference the location can be offered to the
3 individual based on their location. That is, value is provided by the Master World
4 model in the ability to tie services to nodal locations in the Master World.

5 Building on this concept, two additional concepts add value—the concept
6 of so-called Secondary Worlds and a “geozone.”

7 A Secondary World is a powerful computing mechanism whereby
8 individual entities (such as businesses or organizations) can define their own
9 particular worlds that need not necessarily conform to the Master World view of
10 the world. That is, while the Master World is essentially a physical hierarchical
11 representation of the world, the Secondary Worlds can be physical and/or logical
12 representations of each individual entities' world view. One particularly useful
13 aspect of the Secondary World is that it links, at at least one point, into the Master
14 World. Thus, within any Secondary World, a user's location not only within the
15 Secondary World, but the Master World as well can be determined. Various
16 services can be attached to the nodes of the Secondary World. Based upon a user's
17 calculated position, these various services that are associated with Secondary
18 World nodes can be offered to the user. In addition, because the user's context is
19 determined relative to the Master World, other services that may not be associated
20 with a particular Secondary World can be offered.

21 A geozone is essentially a spatial indexing mechanism by which the Master
22 World is subdivided into individual zones. In the described embodiment, the
23 zones are subdivided through the use of a quadtree algorithm that is dependent on
24 a density function (although many other spatial index approaches can also be
25 used). Once a desired density level is achieved (density might be defined in terms

1 of points of interest per zone), each node on the Master World is assigned a
2 particular geozone. Geozones enable proximity calculations to be computed in a
3 fast and straight forward manner.

4 A useful aspect of the Master and Secondary Worlds are that they are
5 "reachable" from various computing devices such as stationary (i.e. desktop
6 devices) or mobile computing devices (i.e. cell phones, laptops etc.). That is, the
7 Master World (or at least a portion of it) and one or more Secondary Worlds can be
8 either locally maintained on the computing device, or accessed, e.g. via the Web or
9 some other mechanism, so that a user can derive their context. For example, the
10 Secondary World can be downloaded onto the computing device so that a user can
11 derive their context within the Secondary World. Once a user's context is
12 determined from the Master World and one or more Secondary Worlds, a various
13 robust collection of context-aware solutions become available to the user. For
14 example, specific Secondary World services can be offered or Master World
15 services can be offered. Additionally, services from other Secondary Worlds might
16 also be offered since the user's location may be known (or made known) to these
17 other Secondary Worlds. In this way, the Master World can link two or more
18 Secondary Worlds together.

19 Another aspect is that the described embodiments harness the computing
20 power of each computing device in determining the device's location. Here, by
21 virtue of having the Master World and one or more Secondary Worlds reachable
22 by the device (and possibly locally maintained on the device), the device itself
23 determines its own context.

24 One embodiment provides a client side device that is configured to utilize
25 the context-aware structures that are discussed above, i.e. the Master and one or

1 more Secondary Worlds. The Master World or a portion thereof can be locally
2 available on the device or can be accessible at another location, e.g. via the Web.
3 In this embodiment, the client device has a location service embodied thereon.
4 The described location service is a software module that can determine the
5 location of the device and can answer queries from various applications (either
6 executing on the device or off the device). The location service determines the
7 location of the device by using the Master World and one or more Secondary
8 Worlds. The applications query the location service through one or more
9 Application Program Interfaces (APIs) or Events to get location information that is
10 used by the applications to render a service.

11 The location service makes use of one or more location providers that
12 convey information to the device. This information can be information that is
13 specific to the location provider, or can be information that can be mapped directly
14 into a node of the Master World or Secondary Worlds. Exemplary location
15 providers can include Global Positioning Service (GPS) providers, cell phone
16 providers (cell providers), Bluetooth providers, a user interface provider and the
17 like. The location providers provide information that gives some aspect of a
18 device's current location. This information is used by the location service to
19 ascertain the location of the device.

20 One particularly advantageous feature of the client device is a standard or
21 common location provider interface. The location provider interface enables the
22 various location providers to provide information to the location service so that the
23 location service can use the information to determine its location. Essentially, the
24 multiple location provider interface is a common interface that enables multiple
25 different location providers to provide location information (or hints) about

1 location to a location service that is on a device. The location providers can
2 provide the location information constantly, at intervals, or when polled by the
3 device. The location information can be provided with confidence and accuracy
4 estimates to enable the location service to evaluate the relative quality of the
5 information before it is used. The various providers also have the ability to self-
6 monitor themselves which assists in the providers' ability to intelligently convey
7 information to the location service. By having a common interface, the collection
8 of location providers is dynamically extensible—that is location providers can be
9 added or removed from the collection of location providers without any
10 interference of the functionality performed by the location service or device. The
11 location providers can be added or removed while the device is operating. This is
12 particularly useful in accommodating location providers that are developed in the
13 future. In this particular embodiment, two levels of abstraction are provided i.e.
14 (1) the provider interface that receives information from the location providers and
15 (2) the API/events layer that enables applications to get at the various information.

16 One focus of this embodiment is a device that can collect context
17 information (e.g. location information) from a variety of different sources,
18 determine the device's current context from that information, and provide the
19 current context at some level to one or more applications that can use the device's
20 context to render a service or enable the device to participate in its context
21 environment.

22 In the described embodiment, the device receives location information or
23 hints about its location. This information is collated and mapped by the location
24 service into a node in the Master World and/or Secondary World. The hierarchical
25 trees can then be traversed to determine the device's accurate location in both the

1 Secondary World and the Master World. At this point, the device has determined
2 its context. The information that is collected can be subject to arbitration to ensure
3 that only highly trusted information is used to determine context. The location
4 information can be cached to provide "current location information" which, for a
5 definable period of time will be accurate to some degree. Thus, if for some reason
6 other location providers are unavailable, the cache can be used to ascertain
7 location.

8 Once a device's location is determined, the device can apply a security
9 policy to the information. Once this is done, the device can begin to answer
10 queries from various applications.

11 One aspect of the described embodiment is a "favorite locations" aspect in
12 which the device can be automatically configured, when it determines its context,
13 so that it can adjust to the different locations.

14 Further, various types of location providers can convey different types of
15 information. For example, a so-called "thin provider" provides location
16 information that is translated by the location service into the appropriate node
17 information. A so-called "thick provider" includes logic that takes location
18 information and provides it in a form that can map directly into the Master World
19 or Secondary World.

20 In another embodiment location translation services are provided that are
21 directed to determining, as accurately as possible, the context or location of the
22 device. In this embodiment, information is received from the various location
23 providers. This information includes location, accuracy and confidence (all of
24 which are provided by the location provider), trust (which is assigned to a location
25 provider by the device or a user) and a timestamp (which helps to age the location

1 information). The location translation processing involves determining which of
2 the location providers are valid and active. The location providers can be ranked
3 in accordance with the confidence and trust levels. This defines an ordered list of
4 location providers. Provision is made for a situation in which all of the location
5 providers may go inactive. If so, a "current location" is used as a location
6 provider whose confidence decreases over time.

7 In the event that information from two or more of the location providers
8 conflicts, then measures can be taken to use information for which there is a
9 higher level of trust. The information that is provided by all of the location
10 providers (assuming no conflict) can then be used to determine a tree structure and
11 a node's entity ID (EID). The tree might be the Master World and the EID is a
12 node on the Master World. The tree might also be a Secondary World and the EID
13 (or location unique identifier or "LUID") is a node on the Secondary World. Once
14 this information is collected, complete location information can be determined by
15 simply traversing the tree(s). Once a device's location is determined, a cache can
16 be updated with the current location (including a time stamp).

17 In another embodiment, privacy issues in the context-aware computing
18 environment are addressed. In this embodiment, the location service has acquired
19 location information that pertains to the location of a particular device. A privacy
20 manager determines what level of information to provide to applications that
21 might request the information. The privacy manager can reside on the computing
22 device itself, or can be proxied by a trusted third party.

23 According to this embodiment, a scale of privacy levels are defined. Each
24 level is defined to include more or less specific information about the location of a
25 particular device. A user is able to assign a privacy level to entities that might

1 request location information. Additionally, each node of the Master World and a
2 Secondary World can have a privacy level associated with it. When a query from
3 an application is received, the privacy manager first determines who the query is
4 from and the privacy level associated with the application or entity. The privacy
5 manager then evaluates one or more of the Master World and the Secondary World
6 to find a node that has a corresponding privacy level. When a corresponding node
7 is found, information at that particular granularity is provided to the requesting
8 application or entity.

9 In another embodiment systems and methods of providing a location
10 provider in the form of a location beacon are described. In this embodiment, a
11 location beacon is provided that can be mounted in various areas (public/private
12 areas) to beacon the location to any computing devices within transmission range.
13 The information that is transmitted enables a device to determine its location or
14 context. The location beacon can transmit information that is specific to the
15 location service that uses the information. Transmitted information can include an
16 EID/URL pair, and a LUID/URL pair. The EID gives the node identification of a
17 node in the Master World; and, the associated URL gives a protocol to
18 communicate with the Master World. The URL might, for instance, link to a
19 server that can provide additional context information that uses the EID. The
20 LUID indicates a node on a Secondary World that corresponds to a current
21 location; and the URL gives a protocol to communicate with the Secondary World.
22 For example, the URL can link with a server that is hosting the Secondary World.
23 This server can then be queried to discover more information about the Secondary
24 World (i.e. Secondary World tree structure, location of associated resources, etc.)
25 With the EID and LUID (along with the URLs), a device can now traverse the

1 Master World or Secondary World to determine its location. Various technologies
2 can be used to implement the beacon (wireless, RF, IR). The beacon can be a
3 “program once” device to deter tampering. Programmable beacons can, however,
4 be provided. Security can also be provided in the form of a verifiable signature
5 that is provided with the beacon information to assure the veracity of the
6 transmitted information.

7 A useful context-aware computing aspect of the beacon is the concept of
8 “location-enabled access”. That is, in addition to (or separately from) receiving
9 location information, a beacon can transmit code download pointers that enable
10 smart devices to access software code that allows the device to participate in its
11 current context.

12 **Exemplary Computing System**

13 In the context of this document, the term “computing device” is used to
14 refer generally to any type of computing device. Characteristics of exemplary
15 computing devices are that they typically include one or more processors,
16 computer-readable media (such as storage devices and memory), and software
17 executing on the one or more processors that cause the processors to implement a
18 programmed functionality. In particular embodiments, implementation takes place
19 in the context of mobile computing devices (e.g. laptop computers and the like),
20 and/or hand-held computing devices (e.g. palm PCs, wireless telephones and the
21 like).
22

23 Fig. 1 is a schematic diagram that constitutes but one example of a
24 computing device that is suitable for use in connection with the described
25 embodiments. It is to be understood that portions of the illustrated computing

1 device can be incorporated in one or more of the computing devices (e.g. palm
2 PCs, wireless telephones, etc.) with which particular embodiments are envisioned
3 for use.

4 Computer 130 includes one or more processors or processing units 132, a
5 system memory 134, and a bus 136 that couples various system components
6 including the system memory 134 to processors 132. The bus 136 represents one
7 or more of any of several types of bus structures, including a memory bus or
8 memory controller, a peripheral bus, an accelerated graphics port, and a processor
9 or local bus using any of a variety of bus architectures. The system memory 134
10 includes read only memory (ROM) 138 and random access memory (RAM) 140.
11 A basic input/output system (BIOS) 142, containing the basic routines that help to
12 transfer information between elements within computer 130, such as during start-
13 up, is stored in ROM 138.

14 Computer 130 further includes a hard disk drive 144 for reading from and
15 writing to a hard disk (not shown), a magnetic disk drive 146 for reading from and
16 writing to a removable magnetic disk 148, and an optical disk drive 150 for
17 reading from or writing to a removable optical disk 152 such as a CD ROM or
18 other optical media. The hard disk drive 144, magnetic disk drive 146, and optical
19 disk drive 150 are connected to the bus 136 by an SCSI interface 154 or some
20 other appropriate interface. The drives and their associated computer-readable
21 media provide nonvolatile storage of computer-readable instructions, data
22 structures, program modules and other data for computer 130. Although the
23 exemplary environment described herein employs a hard disk, a removable
24 magnetic disk 148 and a removable optical disk 152, it should be appreciated by
25 those skilled in the art that other types of computer-readable media which can

1 store data that is accessible by a computer, such as magnetic cassettes, flash
2 memory cards, digital video disks, random access memories (RAMs), read only
3 memories (ROMs), and the like, may also be used in the exemplary operating
4 environment.

5 A number of program modules may be stored on the hard disk 144,
6 magnetic disk 148, optical disk 152, ROM 138, or RAM 140, including an
7 operating system 158, one or more application programs 160, other program
8 modules 162, and program data 164. A user may enter commands and information
9 into computer 130 through input devices such as a keyboard 166 and a pointing
10 device 168. Other input devices (not shown) may include a microphone, joystick,
11 game pad, satellite dish, scanner, or the like. These and other input devices are
12 connected to the processing unit 132 through an interface 170 that is coupled to
13 the bus 136. A monitor 172 or other type of display device is also connected to the
14 bus 136 via an interface, such as a video adapter 174. In addition to the monitor,
15 personal computers typically include other peripheral output devices (not shown)
16 such as speakers and printers.

17 Computer 130 commonly operates in a networked environment using
18 logical connections to one or more remote computers, such as a remote computer
19 176. The remote computer 176 may be another personal computer, a server, a
20 router, a network PC, a peer device or other common network node, and typically
21 includes many or all of the elements described above relative to computer 130,
22 although only a memory storage device 178 has been illustrated in Fig. 1. The
23 logical connections depicted in Fig. 1 include a local area network (LAN) 180 and
24 a wide area network (WAN) 182. Such networking environments are
25

1 commonplace in offices, enterprise-wide computer networks, intranets, and the
2 Internet.

3 When used in a LAN networking environment, computer 130 is connected
4 to the local network 180 through a network interface or adapter 184. When used
5 in a WAN networking environment, computer 130 typically includes a modem 186
6 or other means for establishing communications over the wide area network 182,
7 such as the Internet. The modem 186, which may be internal or external, is
8 connected to the bus 136 via a serial port interface 156. In a networked
9 environment, program modules depicted relative to the personal computer 130, or
10 portions thereof, may be stored in the remote memory storage device. It will be
11 appreciated that the network connections shown are exemplary and other means of
12 establishing a communications link between the computers may be used.

13 Generally, the data processors of computer 130 are programmed by means
14 of instructions stored at different times in the various computer-readable storage
15 media of the computer. Programs and operating systems are typically distributed,
16 for example, on floppy disks or CD-ROMs. From there, they are installed or
17 loaded into the secondary memory of a computer. At execution, they are loaded at
18 least partially into the computer's primary electronic memory. The invention
19 described herein includes these and other various types of computer-readable
20 storage media when such media contain instructions or programs for implementing
21 the steps described below in conjunction with a microprocessor or other data
22 processor. The invention also includes the computer itself when programmed
23 according to the methods and techniques described below.

24 For purposes of illustration, programs and other executable program
25 components such as the operating system are illustrated herein as discrete blocks,

1 although it is recognized that such programs and components reside at various
2 times in different storage components of the computer, and are executed by the
3 data processor(s) of the computer.

4 5 **Defining the World**

6 One of the problems to date with attempting to solve the context-aware
7 computing problem is that every proposed solution has its own approach, data
8 structures, processes and the like. There is little if any standardization between the
9 various approaches. In the described embodiment, standardization is achieved at
10 the foundational level by defining a universal view of the Earth. That is, a
11 universally acceptable definition of the Earth is proposed and is useable in various
12 computing scenarios to enable context-dependent computing. In this document, a
13 specific example of context-dependent computing is given in the form of location
14 dependent computing. It is to be understood that this constitutes but one example
15 of a context in which the various embodiments discussed below can be employed.
16 Other "contexts" can include, any information that can fit into a hierarchical
17 structure including, without limitation, role/personnel in an organization, device
18 categorizations, current activity, current environment, active devices and the like.

19 20 **The Master World**

21 A Master World is defined as a politically correct and publicly accepted
22 hierarchical tree structure that catalogs physical location or geographic divisions
23 of the Earth. The Master World is defined in such a way that many different
24 classes of political, administrative and geographic entities across the entire Earth
25

are included. Areas of political contention are accounted for by presenting a view of the world based on the language/locale of the computing device.

Fig. 2 shows an exemplary hierarchical tree structure 200 that represents a portion of the Master World. The Master World contains multiple nodes 202, with each node representing some type of geographic division (e.g. political or natural entity) of the Earth. In the illustrated example, the nodes of the Master World are arranged in the following groups: (1) political or natural entities (e.g. continents, countries, oceans, states, counties, cities and the like); (2) infrastructure entities (e.g. postal codes, area codes, time zones and the like); (3) public place entities (e.g. parks, malls, airports, stadiums, and the like); and (4) non-physical entities (military postal code regions, vacation regions, affiliate coverage areas of television networks that can be geographically discontinuous, and the like).

In the Fig. 2 example, the top node of the tree structure represents the Earth. Each node underneath the top node represents a geographical division of the Earth. In this example, none of the nodes have an association with any businesses or services. That is, there is a distinction between node entities that are part of the Master World and non-geographic places where activities take place. Though the Master World includes nodes for public places (i.e. airports, malls, etc), it does not include individual listings of businesses or service providers. Each node is uniquely identified by an ID (EID or entity ID). In addition to the unique EIDs, a URL is associated with the tree structure and provides a context for the tree structure as will become apparent below.

As an example, consider the following: Seattle-Tacoma International Airport (SeaTac) will be included in the Master World, but references to individual airline business locations at SeaTac might be "leaves" on the tree that are tagged

1 by the SeaTac Airport EID (see "Secondary World" section and the Table below).
2 Similarly, the Seattle Center might be a node on the Master World, while the
3 Seattle Arts Festival, Bumpershoot, the Seattle Sonics NBA Team, and the Seattle
4 Center Starbucks Coffee Shop might be tagged with the Seattle Center EID. As
5 another example, the Master World also contains nodes for all Interstate
6 (motorway) exits. For example, the I-90, Exit 109, Washington is a node in the
7 Master World. The Best Western Inn located at 1700 Canyon Road in Ellensburg,
8 Washington might be tagged with the EID of this Exit.

9 Thus, the Master World provides a uniform way of defining locations. The
10 uniform location definitions can then be universally used to assign attributes to
11 goods or services. Whenever a computing device determines its location to
12 correspond to a particular uniform location definition, it can take advantage of the
13 location-dependent goods or services that share the uniform location definition.
14 The Master World is useful because it is a standardized view of the world. Its
15 accurate standardized geographic dimension attribution can be easily accessed by
16 both providers and consumers. Services and product providers (or third parties
17 such as search engines, network and yellow-page database directories) can use the
18 nodes of the Master World by assigning a standardized persistent geographic
19 reference to all commerce locations or points of interest. These commerce
20 locations or points of interest can be considered as "leaves" on the tree structure.

21 In the illustrated example, the nodes of the Master World have one or more
22 attributes that facilitate its use. Exemplary attributes are described in the table
23 immediately below:
24
25

Attribute	Description
Entity ID (EID)	The EID is a unique ID for each node. No two nodes have the same EID.
Name	The name is defined in terms of the neutral ground truth (NGT) name. The NGT name supports various language translations for entity names as appropriate (e.g. Pacific Ocean, Pazifischer Ozean, Oceano Pacifico, etc.)
Geographical Entity Class (GEC)	The GEC is a geographical classification of each node. An exemplary GEC is discussed below in the "Geozone" section.
Latitude	The horizontal coordinate position on the globe (i.e. the coordinate position of the node's centroid)
Longitude	The vertical coordinate position on the globe (i.e. the coordinate position of the node's centroid)
Relative Importance	The geographic importance of an entity in reference to other entities in the same region. Value from 1 to 256 (e.g. New York City = 3, Los Angeles = 4, and Omaha = 5 even though Omaha is much smaller but almost as important in relation to surrounding populated places)
Contextual Parent(s)	The parents of the parent/child relationship for each node. Multiple parents are supported (e.g. Redmond is a child of King County, Area Code 425, the Pacific Time Zone, and the MSNBC affiliate KING TV).
Source	The source of origin for the record (e.g. Microsoft or a specified data vendor)
Start Date	Date when the node information was first valid
End Date	Date when the node information was last valid (retired zip codes, breakup of countries)
Modification Date	Records date changes that are made to the record relating to retirement or updates to any fields
Status	Active, lashed (links duplicate nodes together), pending or retired

The attributes listed above constitute exemplary attributes only. Other attributes that are different from and/or additional to those referenced above could be used. A few exemplary entity or node records that employ the above attributes are shown below:

Entity ID (EID)	24948
Name	Pacific Ocean, Pazifischer Ozean, Oceano Pacifico, etc.
Geographical Entity Class (GEC)	138/Ocean
Latitude	0 (+000° 00' 00")
Longitude	-170 (-170° 00' 00")
Relative Importance	1
Contextual Parent(s)	World
Source	MSFT GeoUnit
Start Date	0/0/00

End Date	0/0/00
Modification Date	01/18/00
Status	Active

Entity ID (EID)	27490
Name	Redmond
Geographical Entity Class (GEC)	78/non-capital town
Latitude	47.6768303 (+047° 40' 36")
Longitude	-122.1099625 (-122° 06' 35")
Relative Importance	107
Contextual Parent(s)	<ol style="list-style-type: none"> 1. King, second level [Washington, United States] 2. Puget Sound-Seattle, travel region [Washington, United States]
Source	MSFT GeoUnit
Start Date	0/0/00
End Date	0/0/00
Modification Date	01/18/00
Status	Active

The Master World also serves as a repository of common denominator links between itself and various "Secondary Worlds" and as a conduit that connects Secondary Worlds to other Secondary Worlds. Content, service and device providers can use the Master World to associate their publicly available offerings with a geographic location and the corresponding multiple branch hierarchical structure. This location will be associated with a single entity within the tree structure thereby allowing geographic and time/distance calculations and the necessary parent/child relationship navigation.

The Master World Index (Geozones)

By definition, the Master World provides a hierarchical structure of entities (nodes) that cover the entire globe. Upward navigation within the hierarchy is quite natural. Efficient navigation downward requires geographic proximity awareness. Additionally, there are possible scenarios that will require jumping from branch to branch in order to successfully return values in a query, or for more accurate calculations of distances to close "leaves" attached to nodes other than the original source node. The Master World makes use of an index scheme that can identify peer level nodes by virtue of the geographical proximity. This indexing scheme makes use of a quad tree algorithm to define so-called "geozones."

A quadtree is essentially a spatial index that breaks coverage into homogeneous cells of regularly decreasing size. Each quadrant of the tree has up to four children. The quadtree segmentation process can continue until the entire map is partitioned based on many different end result criteria including the density of the number of items (e.g. points of interest) in each quad. The approach provides a form of spatial index that accelerates spatial selection and content identification.

To complete the spatial indexing scheme to provide each node with a defined geozone, a quadtree algorithm is applied to the nodes and can be based upon a desired density of, for example, points of interest that are to occur in any one zone. Once all of the zones have been defined, each zone is given a unique ID (e.g. top/left and bottom/right Latitude and Longitude pairs). Each of the nodes of the Master World is then assigned a zone in which it is located. Quadtree algorithms are known and will be appreciated by those of skill in the art.

The Master World Location

As can be appreciated, having a uniform standardized representation of the world in the form of a hierarchical traversable tree structure can greatly facilitate the manner to which context-dependent, and more specifically, location-dependent goods and services can be linked.

In the described embodiment, a computing device has access to at least a portion of the Master World. For example, the computing device can have the Master World saved in an internal storage device, it can comprise part of the computing device's operating system, or the device might access the Master World via a network medium such as the Internet. With the Master World tree structure being accessible to each computing device, each device has the power to determine its own context or node-referenced location. That is, the computing device can determine, through software it is executing, its particular location, i.e. node. Once the computing device determines an associated node, it can simply traverse the tree to ascertain its complete location.

For example, if a computing device determines that it is currently located at a node that corresponds to the City of Redmond, it can traverse the Master World tree structure to ascertain that it is in the State of Washington, Country of The United States, on the continent of North America. By ascertaining its precise location, the computing device (or its user) is now in a position to take advantage of location-dependent services that might be offered. This particular model is a tremendous improvement over current models that utilize a central server to ascertain location for a number of different devices. In that model, each device (or user) provides information about its location (e.g. perhaps the user enters the zip

code or city that the device is currently in) and might enter a query to find, for example, a McDonald's restaurant in his zip code. The server then takes this information and might, for example, tell the user about the location of all of the McDonald's restaurants within that zip code or city. If the servers fails in this model, then none of the computing devices can take advantage of its services. In the present model, each computing device is self-sustaining. Each can determine its own location, and accordingly, each device can take advantage of location-dependent services. For example, if the computing device understands that it is located on a particular node of the Master World, then it can execute queries to find a McDonald's that has an EID that corresponds to the particular node in which the computing device is located. Particular robustness is provided through the use of the above-described geo-zones. The geo-zones enable proximate geographic divisions to be quickly searched in an efficient manner. For example, if an individual is looking for the nearest Kinko's to make copies and none are located in the geo-zone that corresponds to the node in which the computing device is located, then adjacent geo-zones can be quickly searched.

Secondary Worlds

In the described embodiment, the concept of a Secondary World is used to provide support for additional context. A secondary world might be defined by a third party organization or company and contains nodes that comprise physical and/or logical entities that are unique to that organization. The nodes of the Secondary World may or may not have much context outside of the particular organization that defined the Secondary World, since a secondary world could be made either public or private. The Secondary Worlds do not duplicate the Master

World, but rather supplement it in a unique, organization specific manner. While the Master World is defined to be a widely accepted standard, each Secondary World can be a widely variant representation of an organization's proprietary view of the world. In the described embodiment, each Secondary World has at least one node that is linked with a node of the Master World. This gives the Secondary World a context or location in the Master World. Also note that in some context applications, several secondary worlds may be accessed, each providing additional context specific pieces of location data.

Fig. 2 shows an exemplary Secondary World 204 that comprises a plurality of nodes 206. Each of the nodes 206 constitutes a physical or logical entity. For example, the nodes can constitute a company, its divisions, regions campuses, buildings, floors in various buildings and rooms on various floors. At least one of the nodes is linked with a node of the Master World. The nodes of the Secondary World can have the same attributes as the nodes of the Master World.

As an example of a Secondary World, consider that Boeing might define a Secondary World that includes a list of entities that are important to its employees. The root entity would be "Boeing Corp." and its children might be company divisions (St. Louis Military Division, Everett Plant, Corporate HQ, etc.). Further down the tree structure, individual nodes might be defined to represent individual buildings (Hanger 12), offices within this building (Office 1001), building areas (Southwestern quadrant of hanger 12), etc. Each entity or node has a unique identifier (Local Unique ID or "LUID") and a URL that is associated with the tree on which the node occurs. The URL uniquely identifies the Secondary World tree structure so that a user within that world can determine how to interact with the world. This aspect is discussed below in more detail. Boeing can then use the

1 LUIDs to associate equipment, services, departments or even personnel to a
2 physical or logical location.

3 As a more concrete example, consider Fig. 3 which shows an exemplary
4 portion of the Master World 300 and a Secondary World 302. Master World 300
5 includes the following nodes: World, United States, Washington, Redmond, and
6 Zip = 98052. The exemplary Secondary World 302 is a hierarchical tree structure
7 that has been defined by Microsoft Corporation and includes the following nodes:
8 Microsoft, Redmond Campus, 1 Microsoft Way, Building 26, 3rd floor, Conference
9 Room 3173, Building 24, 2nd floor, Conference Room 1342. In this example, the
10 Secondary World 302 “touch points” into the Master World from the Redmond
11 node. In this example, a video projector is shown as being associated with the
12 node “Conference room 1342”. Here, the video projector is not a node in the
13 secondary world. Rather, the video projector is an item in some other resource
14 discovery service (e.g. the active directory) and includes a location attribute that is
15 a pointer to “Conference room 1342.” There may be times, however, when nodes
16 can be created in the worlds to represent the location of key services—the node
17 themselves, however, would not represent the services.

18 Like the Master World, the Secondary World is advantageously accessible
19 to a user’s computing device. It could, for example, be downloaded—completely
20 or partially--and stored on a storage device and accessed when needed. It might
21 be downloaded for a one time use only. The Secondary World enables the
22 computing device to ascertain its context within the Secondary World. In this
23 example, the computing device would, by using the Secondary World, compute its
24 location within the Secondary World. The computing device can do this by
25 traversing the tree structure from the node in which it is currently located to the

1 root node. This would, for example, give the computing device (and hence the
2 user) a complete Secondary World context. Once the Secondary World location is
3 known, the user is in a position to take advantage of goods or services that are
4 associated with the nodes of the Secondary World. That is, once the computing
5 device determines its Secondary World context, it is ready to become an active
6 participant in the Secondary World.

7 Tremendous value can be achieved by associating goods or services with
8 the individual nodes of the Secondary World. For example, Conference Room
9 1342 has a video projector associated with it. That is, the location of the video
10 projector is in Conference Room 1342. Assume that an individual in Conference
11 Room 3173 has a presentation that requires the use of the video projector such as
12 the one located in Conference Room 1342. Normally, an individual would have
13 no way of ascertaining the location of the video projector other than perhaps
14 physically calling over to the building to check whether there is a video projector
15 available. In this example, because the user's computing device is able to
16 ascertain its location within the Secondary World, it is able to locate the video
17 projector in Conference Room 1342. It would do this by simply executing
18 software that traverses the Secondary World tree structure to find the resource of
19 interest.

20 Note also that because there is a link into the Master World, the computing
21 device is able to derive its context (location) within both worlds. This enables the
22 computing device, and hence the user, to take advantage of goods and services that
23 are associated with the Secondary World, as well as participate in location-
24 dependent services that are consumable based upon the user's location in the
25 Master World.

1 Fig. 4 is a flow diagram that describes steps in a method in accordance with
2 the described embodiment. The steps described just below are implemented by a
3 computing device which, in the illustrated example, is a hand-held mobile
4 computing device.

5 Step 400 accesses first and second hierarchical tree structures that are
6 resident on a computer-readable media. In this example, the tree structures might
7 be stored on the device or might be accessible via a network such as the Internet.
8 An exemplary first tree structure is the Master World and an exemplary second
9 tree structure is a Secondary World. Alternately, the tree structures could both be
10 Secondary Worlds. Once the tree structures have been accessed by the device,
11 step 402 traverses multiple nodes of the tree structures to derive the context of the
12 computing device. In this example, the computing device receives information
13 that informs it as to its location at a node of one of the trees. This information can
14 come to the computing device in any suitable way, e.g. a user can enter the
15 information through a User Interface (UI) or the location might be broadcast to the
16 computing device by another computing device (e.g. through the use of Bluetooth
17 technology or Universal Plug and Play (UpnP). Specific examples of how this
18 information can be conveyed to the computing device are given below in more
19 detail. Regardless of how this information is conveyed to the computing device,
20 once the computing device has the information, it executes software that traverses
21 one or both of the tree structures to derive its context which, in this example, is the
22 device's location.

23
24
25

Defining Secondary Worlds

As was mentioned above, one particularly valuable aspect of the described embodiment is that individual organizations can define their own Secondary Worlds. This gives the organization a great deal of flexibility in providing goods and services and, more broadly, increasing the efficiency of their organization. In one embodiment, a software tool is provided that enables individual organizations to define and maintain their own Secondary Worlds.

In one embodiment, each secondary world can be uniquely identified as a name space (e.g. an XML namespace). This ensures that any overlap in names between the Secondary World and the Master World will not result in a collision. As an example, consider the following: the Master World might contain an entity identified as "Chicago" referring the city. A Secondary World that is established by the National Basketball Association (NBA) and a different Secondary World that is established by the Caterpillar Corporation might also have entities named "Chicago" that refer to completely different entities than the Master World's "Chicago." For example, the NBA's "Chicago" might refer to an NBA market area while Caterpillar's "Chicago" might refer to a sales district. Having the namespace separation between the Master and Secondary Worlds can ensure that there not a collision between identically named entities because each name space is uniquely different from every other namespace.

Fig. 5 is a flow diagram that describes steps in a method of building a context-aware data structure. These steps are implemented by a software tool that is executing on a computing device.

Step 500 receives input from a source that specifies information that pertains to physical and/or logical entities. In this example, a system administrator

might physically enter information about the structure of the Secondary World that they desire to define. This information can include information about buildings, divisions, conference rooms and the like. Step 502 then processes the information to define a hierarchical tree structure that has a context. In this example, the context is location. It will be appreciated, however, that other contexts could be employed. Each of the nodes in the hierarchical tree structure represents a separate physical or logical entity. Step 504 then links at least one of the nodes of the hierarchical tree structure with another tree structure having a context. In this example, this other tree structure can comprise the Master World. Once the tree structures have been built and linked, they are ready for traversal in a manner that enables context to be derived from one or more of the nodes.

Location as a Service

In the above examples, the computing device is able to determine its own location. In the embodiment about to be described, the computing device determines its location by using location information that is provided to it from a number of different sources of information. The device is able to take the information that is provided to it and process the information to determine a particular node on one or more hierarchical trees. Once the device has done this, it can determine its complete location which is a useful thing to know particularly when there are location-dependent services that can be consumed by the device's user.

Fig. 6 shows a high level diagram of an exemplary computing device 600 that comprises, among other components, a context service module 602 and one or more context providers 604. The context service module 602 can be implemented

1 in any suitable hardware, software, firmware or combination thereof. In this
2 particular example, the context service module is implemented in software that is
3 executed by one or more device processors. The context service module 602
4 receives context information from one or more context providers 604 and
5 processes the information to determine a current device context. In this particular
6 example, the device context is the device's location. Accordingly, the context
7 providers are location providers that provide location information, in various
8 forms, to the context service module 602 for processing. The location providers
9 604 receive information from various sources of context information (location
10 information) 606.

11 In the context of this document, a context provider comprises a software
12 component that can either be implemented on the device or off the device. The
13 context provider can also include any suitable hardware, firmware or combination
14 thereof. The role of the context providers are to receive information from sources
15 606 and convey the information to the context service module 602 so that the
16 context service module can use the information to determine a current device
17 context.

18 In the case where the context of the device is the device's location, sources
19 606 provide various information to the location providers 604 that pertains to the
20 device's current location. As an example, the sources of the information can
21 include various information transmitters such as a GPS system, cell phone or cell
22 ID, wireless transmitters that transmit location information, location beacons,
23 802.11 transmitters and various other sources of information. The sources of
24 information can also include other computing devices that might, for example,
25 provide location information through Bluetooth technology. In addition, a source

1 of information 606 might include a person who, for example, physically enters
2 location information into the device 600 so that the device can process the
3 information to determine its location.

4 When the device 600 receives the location information from the sources
5 606, it processes the information with the location providers 604 and provides the
6 information to the location service module 602. The location service module 602
7 processes the location information and determines a particular node on one or
8 more of the hierarchical tree structures to which it has access which corresponds to
9 its current location. The location service module 602 can then traverse the tree
10 structures to determine a complete location for the device. Once the complete
11 location is determined, the device 600 can begin to interact with one or more
12 applications 608 that can query the device about its particular location so that one
13 or more location-dependent services can be rendered to the device. In this
14 example, the applications 608 are illustrated as being separate from the device. It
15 is to be understood, however, that the applications could be executing on the
16 device, e.g. a browser application.

17 As shown, the applications 608 can make calls to the device to ask the
18 device where it is located. The device is configured to receive the calls and
19 respond in an appropriate manner to the application. Once the application has the
20 device's location information, it can then render location specific services to the
21 device.

22 **Consider the following example:** You are a traveler and have a hand-held
23 mobile computing device that contains a Master World tree and a Secondary
24 World tree for SeaTac International Airport. You are scheduled to depart on a
25 plane for China from Concourse C. SeaTac International Airport has designed its

1 search engine application looks for a Double Tree hotel that has an attribute that
2 includes an EID that matches your EID. If it finds one, it simply indicates for you
3 the result. If it does not find one with the corresponding EID, it can use an
4 adjacent geozone to search for a Double Tree hotel. It may also provide driving
5 directions to the hotel. The search engine application was able to do this because
6 it was able to ascertain your location in the Master World. It did this quickly and
7 automatically with little or no effort from you.

8 Consider further that as you are driving from the airport to the hotel you
9 decide that you want to find the nearest Kinko's so that you can print 100 copies of
10 a presentation that you are to give in the morning. Consider that your hand-held
11 computing device is a cellular phone and that Sprint is the carrier. Sprint has
12 defined its own Secondary World that might, for example, be designated in terms
13 of cell nets. By virtue of having Sprint's Secondary World on your computing
14 device, you are able to ascertain your location in Sprint's Secondary World and,
15 accordingly, your location in the Master World. Consider that Kinko's also has a
16 Secondary World that links with the Master World. By executing a search
17 application on your device, you are able to ascertain the location of the nearest
18 Kinko's as well as driving directions thereto. All of this is possible because your
19 device has access to the Master World and one or more Secondary Worlds. In this
20 example, the Master World provides a mechanism to daisy chain two or more
21 Secondary Worlds together. This is possible because the Secondary Worlds have
22 at least one reference or link into the Master World.

Exemplary Device Architecture

Fig. 7 shows computing device 600 in somewhat more detail. In this particular embodiment, device 600 comprises an architecture that includes the following components: a location service module 602, a location provider interface 700, an application program interface (API)/Events module 702, a privacy manager 704 a location conversion module 706, one or more applications 608 and one or more location providers 606. Also included in the architecture is an active directory 708, Web service 710, location database 712, and personal places 714. The architecture can be implemented in any suitable hardware, software, firmware or combination thereof. The architecture mentioned above is advantageous in that it enables each computing device to determine its own context or location.

Common Location Provider Interface

One particularly advantageous aspect of the described embodiment is that it employs a common interface 700 that provides a standard interface through which the location providers 606 communicate. By having a common interface, the location providers are extensible (to support future providers) in that they can be dynamically added or removed from the collection of location providers. All that is required of a particular location provider 606 is that it be written to support the common interface.

In this example, there are several location providers 606. These location providers provide location information in different forms. For example, a GPS location provider might provide location information that is GPS specific. Similarly, an IP/Subnet location provider might provide information that is

specific to an Internet Protocol. A mobile phone location provider might provide location information in the form of a cell ID. In addition, a location User Interface (UI) might provide location information in the form of a user entry that specifies a city, street or building. All of the location information that is provided by the various location providers is processed by the location service module 602 so that a current device location can be determined. To determine the current device location, the location service module 602 may have to consult with an active directory 708, a Web service 710, or a location database 712. In the illustrated example, the active directory 708 might, for example, maintain a secondary world and other networking metadata such as subnet and "site" information that can help determine location based on networking connectivity. Web service 710 can hold the master or secondary worlds, the attributes of which can be used to find location. For example, if a cell phone knows its cell tower ID, then the location provider can query the secondary world to ascertain the nodes that match that cell tower ID. Location database 712 is basically a version of the web service that is hosted or cached locally.

Location Providers

As indicate above, the architecture contemplates multiple different location providers that can provide location information to the location service module 602. This information can come in many different forms and quality levels. The information is then processed by the location service module 602 to determine a current device location. To do this, the service module 602 ascertains from the location information a particular node ID (EID and/or LUID) and a URL that is associated with the tree structure with which the node is associated. Once the

1 location service module ascertains a node ID, it can then query the tree structure
2 (or more accurately a server that manages the tree structure) using the URL to
3 ascertain more information about the tree structure. For example, if the location
4 service module 602 ascertains a LUID from a particular Secondary World, it might
5 then query an active directory 708 (or an Intranet server—which is another
6 location database) to discover the parents and the children of the node. This would
7 then enable the location service module to build the Secondary World.

8 The location providers 606 can provide the location information to the
9 location service module 602 in many different ways. For example, some location
10 providers 606 may continuously provide information (e.g. the GPS provider may
11 continuously provide GPS coordinates). Alternately, the location providers can
12 periodically provide location information such as at specific times or on the
13 occurrence of definable events. For example, a user may define specific times
14 when the location information should be updated. Alternately, the location
15 information might be updated only when a device's location changes (i.e. a
16 location change event). Additionally, the location providers might provide
17 location information when polled by the location service module 602. For
18 example, the location service module 602 can call the location provider interface
19 700 and request location information from one or more of the location providers.

20 One specific location provider 606 is shown as a cache. The cache provider
21 essentially maintains a current device context or location. That is, once the
22 location service module 602 has ascertained its current location, it writes this
23 location to a cache. This enables the device 600 to ascertain its location with a
24 degree of confidence in the event all of the other location providers are not able to
25 provide location information (e.g. the GPS provider may not receive GPS

1 information because the GPS transmitter that supplies it with the information is
2 unable to contact a requisite number of satellites).

3 4 **Confidence and Accuracy Parameters**

5 One important and useful feature of the described embodiment is that one
6 or more of the location providers are configured to assign confidence parameters
7 and/or accuracy parameters to the information that they provide to the location
8 service module 602. Confidence parameters provide a measure of a provider's
9 confidence in the information that it provides to the location service module 602.
10 For example, assume that a GPS transmitter must receive information from five or
11 more satellites in order to provide highly confident information. Assume that only
12 three satellites are available at the time. The GPS transmitter would then transmit
13 its information based only on the three satellites. The GPS provider would then
14 know that the information it receives from the GPS transmitter was based only on
15 three satellites rather than the desired five or more. In this case, the GPS provider
16 can set a confidence parameter on the location information that indicates that it has
17 a lower confidence level than if the information were based on the desired five or
18 more satellites. In this case, the location service module 602 can take the
19 confidence parameters for all of the location providers into account when
20 determining the location of the device. This is discussed in more detail below.

21 With respect to the accuracy parameters, consider that the location
22 information that is received from the location providers is accurate to varying
23 degrees. Some information may be accurate to within one mile, while other
24 information may be accurate to within 100 feet. The location providers are
25 desirably configured to assign accuracy parameters to the location information that

1 they provide to the location service module 602. The accuracy parameters give
2 the location service module an indication of the accuracy of the information.

3 When the confidence and accuracy parameters are used by the location
4 service module 602, the module can make decisions on how to use the location
5 information it receives from each provider. For example, the location service
6 module 602 might disregard completely any information that has a low confidence
7 parameter. It might, on the other hand, strike a balance between the accuracy of
8 the information and its confidence. For example, the module 602 might be
9 programmed to use information with lower levels of accuracy only when there is a
10 high level of confidence in the information. The module 602 might utilize the
11 parameters to assign weights to the information so that the location is calculated as
12 a weighted function of the confidence and accuracy of the information.

13 Another use of the confidence parameters is as follows: Assume that the
14 location service module has determined a device location and has written that
15 location to a cache. At the time when the location is written to a cache, it is
16 assigned perhaps a high confidence level. Assume further that all of the other
17 location providers are unavailable to provide location information. For a period of
18 time, the location service module 602 can use the cache location as a current
19 location and be fairly confident that its information is generally accurate. In this
20 case, the location service module might assign a linearly decreasing confidence
21 level to the information over time so that at some point, it ceases to use the
22 information or informs the user that the information cannot be guaranteed.

Location, Trust, and Timestamp

When the location providers provide their information to the location service module 602, the information can include, in addition to the confidence and accuracy parameters, the actual location information in a known format, a trust parameter and a timestamp. The trust parameter is a metric that is assigned by the location service module 602 to one or more of the location providers and defines the trust that the location service module has for the particular location provider. The timestamp is a metric that defines the time when the location information was provided by the location provider. This assists the location service module 602 in ascertaining whether information is stale and might need refreshed.

Once the location service module 602 has all of the location information, it can then set about determining the location of the device.

Fig. 8 is a flow diagram that describes steps in a method of determining a device context which, in this example, is the device location. These steps are implemented by the location service module 602.

Step 800 gets the current device context. The current context can be the last calculated device context that is stored in the cache. Step 802 determines whether any of a number of context providers are available to provide context information. The location service module might do this by polling the context providers to ascertain which of the providers are active and valid. Step 804 determines whether all of the providers are inactive. If all of the providers are inactive, step 806 decreases the confidence in the current context over time and uses the current context as the device context. Step 802 then continues to monitor for current active and valid providers. If step 804 determines that one or more of the context providers are active, then step 808 orders the active and valid context

1 providers. When the location service module 602 orders or sorts the context
2 providers, it does so as a function of the confidence of the provider's information
3 and/or the trust that the location service module has in the location provider. This
4 provides a ranked list of the location providers. Step 810 checks to ascertain
5 whether the context information appears to be correct. For example, where the
6 context is the location of the device, the location service module 602 might know
7 that five seconds ago the current location was Redmond, Washington.
8 Accordingly, location information that indicates that the current location is
9 Beijing, China would be incorrect. Step 812 then determines whether any of the
10 context information conflicts with either the device's current context or the context
11 information from other providers. For example, the location service module 602
12 can compare the context information from each of the context providers with the
13 information in the cache. If any of the information conflicts with the cached
14 information, then the information from that context provider can be discarded.
15 Similarly, if context information varies inordinately as between the context
16 providers, then step 814 can select the context providers having a predefined level
17 of trust and perhaps use just their information (Step 816). If there are no conflicts,
18 then step 816 determines the current context based upon the information that is
19 provided by all of the context providers. In the described embodiment, this step is
20 implemented by using the information to map to a particular node in one or more
21 of the hierarchical tree structures mentioned above. For example, the location of
22 the device can be ascertained by mapping the information to a particular node, and
23 then completely traversing the tree structure until the root node is reached. Step
24 818 then updates the current context by perhaps writing it to the cache and returns
25 to step 802 to determine the active and valid context providers.

1 The method described above provides a way for the location service
2 module to receive location information and use only the location information that
3 appears mostly likely to represent a current location. Conflicting information can
4 be discounted or disregarded thereby assuring that only the most trusted, accurate
5 and confident information is utilized to determine the device's current location.

6 7 **Self Monitoring**

8 In addition to the confidence and accuracy parameters, one or more of the
9 location providers are advantageously programmed to self monitor their own
10 operation for various irregularities that can occur. On the occurrence of an
11 irregularity, the location providers are configured to notify the location service
12 module 602. For example, the source from which the location provider receives
13 its information may go off line for a period of time so that the location provider is
14 unable to receive any additional information. In this case, the location provider
15 might generate a "provider out" message and send it to the location service
16 module 602. When the location service module 602 receives the "provider out"
17 message, it can then take steps to exclude the location information from that
18 provider from any location calculations that it performs. When the location
19 provider's source comes back on line, it can generate a "provider on" message that
20 informs the location service module 602 that it is able to transmit location
21 information to the module. Of course, the location service module can be notified
22 by the location providers on the occurrence of other operational irregularities, with
23 the above example constituting but one specific case.

Applications

Once the location service module 602 has determined the device's location, it can receive queries from one or more applications 608. In the Fig. 7 example, the applications include a web site application, an Outlook application, and a service discovery application. In the present example, the web site application can be any web site application that is capable of rendering location-specific services. For example, the user of the device 602 might access Amazon.com's web site to buy a favorite book. When the user purchases their book, Amazon.com must now compute the taxes that the user must pay. In this example, a script executing on Amazon.com's web site might query device 602 to learn of the user's location. In this particular example, the device might respond to the query by returning the state in which the user is making the purchase. Amazon.com can then assess the tax automatically. Amazon.com might also desire to know where the individual is located so that they can select an optimal shipping method (UPS or Express Mail). Depending on where the individual is located, one method may be preferred over the other. The Outlook application might query the location service module to ascertain the location because it (or the operating system, e.g. Windows) may change device settings based on the location of the computing device. For example, the user may print on one particular printer while at work, and another particular printer when at home. When the Outlook application determines that the user has gone home for the day, it can automatically change the device settings for the printer at the user's home. It might acquire the print settings from a personal places data store 714. Thus, the device is automatically configured for use depending on the user's location. The service discovery application might query the device to determine its location so that it can render a particular service

1 depending on where the device is located. For example, if the user asks the
2 application to locate the nearest color printer, the service discovery application
3 might query the location service module to ascertain the device's current location
4 so that it can use this information and find the nearest color printer. Consider also
5 that the Outlook application could configure itself email to a work location (when
6 an individual is at work) or to a home location (when an individual is at home). In
7 addition, the Outlook calendar can become location aware, e.g. when you change
8 time zones, your appointments would show up in the proper time slots.

9 As one can imagine, the possibilities are seemingly endless. This
10 functionality is made possible through the use of the Master World and one or
11 more Secondary Worlds.

12 13 **Application Program Interface/Events**

14 In the described embodiment, the applications 608 communicate with the
15 location service module 602 through one or more application program interfaces
16 (APIs) and/or events. The applications can make function calls on the API to
17 query the location service module as to its current location. Similarly, the
18 applications can register for location notifications by using an events registration
19 process. For example, an application may register for a notification when the user
20 changes their location. Consider the case where an application requests to be
21 notified when the user arrives at work or at home so that the application can
22 change the device's configuration (such as printer configuration).

23 Fig. 9 is a flow diagram that describes steps in a method in accordance with
24 the described embodiment. The steps that are described are implemented by
25 device 600. Step 900 receives information that pertains to the current context of

1 the device. In this particular example, a portion of the information is received
2 from one or more context providers which, in this case, are location providers.
3 Step 902 processes the information on and with the device to ascertain the current
4 context of the device. In the illustrated example, the device maintains (or has
5 access to) one or more of the Master World and one or more Secondary Worlds.
6 When the device receives all of the location information, it maps the information
7 to a particular node in the hierarchical tree structure that defines the Worlds. It
8 then traverses the tree structures to ascertain the complete context (i.e. location) of
9 the device. Step 904 receives calls from one or more applications that request
10 information that pertains to the device's current context or location. In the
11 illustrated example, the applications can call one or more APIs to request the
12 information or the applications can register for event notifications. Step 906 then
13 supplies the applications with at least some information that pertains to the current
14 device location. As will be discussed below, a security policy or privacy policy
15 can be applied to the information before it is returned to the applications.

16 17 **Privacy Manager**

18 In one embodiment, a privacy manager 704 (Fig. 7) is provided. Although
19 the privacy manager is illustrated as being incorporated on the device, it could be
20 implemented by a trusted entity such as a trusted server that is not part of the
21 mobile computing device. The privacy manager can be implemented in any
22 suitable hardware, software, firmware or combination thereof. In the illustrated
23 example, the privacy manager comprises a software module that is incorporated in
24 the mobile computing device.
25

The privacy manager 704 addresses privacy concerns that are associated with the information that is collected by the computing device. Specifically, the location service module can calculate detailed information regarding the location of the computing device. It may be desirable, in some instances, to filter the information that is provided to various applications. That is, it is entirely likely that a user may not want their specific location information provided to untrusted applications. In these instances a user might just desire for location service module 602 to inform such applications that the user is in the State of Washington.

Fig. 10 shows a flow diagram that describes steps in a privacy protection method in accordance with the described embodiment. These steps can be implemented by the privacy manager 704.

Step 1000 defines a plurality of privacy levels. Exemplary privacy levels are set forth in the table immediately below:

Privacy Level	Approximate Scale	Level of Revelation
0	-	No location information is returned
10	100,000 Km	Planet/Continent
20	1,000 Km	Country
30	100 Km	State
40	10-100 Km	City & County or Region
50	10 Km	Postal Code & Phone Area Code
60	1 Km	Full Postal Code (Zip + 4) & Area Code and Exchange
70	100 m	Phone Number & Building/Floor

80	10m	Room #
90	1m	Exact Coordinates

In the illustrated table, 10 different privacy levels are defined and each has an associated approximate scale. For example, a privacy level of 0 means that no location information is returned. A privacy level of 90 means that very detailed location information is returned.

Step 1002 assigns various privacy levels to the individual nodes in one or more hierarchical tree structures. For example, each node of the Master World and the Secondary Worlds can have a privacy level associated with it. The root node of the Master World tree structure might have a privacy level of 10, while the node that represents a current location in a Secondary World might have a privacy level of 90. Step 1004 determines the context of the computing device. In the present example, the context is the device location and examples of how this is done are given above. Individual applications that call the location service module can have privacy levels associated with them. These privacy levels can be assigned by individual users. For example, a trusted application might have a privacy level of 90, while an untrusted application might have a privacy level of 30. Step 1006 receives context queries from one or more applications. Here, an application calls the location service module 602 (Fig. 7) to ascertain the location of the device. Step 1008 determines the privacy level associated with the application or applications. For example, if a untrusted application calls to request location information, the privacy manager 704 would determine that the application has a privacy level of 30. The privacy manager then traverses (step 1010) one or more hierarchical tree structures to find a node with a corresponding privacy level so

that it can select the information that is associated with that node. In this example, the traversal might involve jumping from the Secondary World to the Master World to find the node that corresponds to the state in which the user is located. Once the corresponding node is found, step 1012 returns the context information (e.g. location information) associated with the node. In this case, the location service module would inform the application that the user's location is the State of Washington.

As an example, consider the following: There is a web site that gives up to the minute weather of various locations. Accordingly, you might assign this web site a privacy level of 60 so that you can receive weather information for the geographical area that corresponds to your present full postal code. Another web site might be a corporation intranet web site that is a trusted web site. Thus, any applications associated with this web site can be assigned a privacy level of 90 so that you can give them precise location information as to your whereabouts.

Thus, in the present example, the computing device is able to determine the source (i.e. application) of its queries and modulate the information that is returned to the application as a function of the application's identity. The computing device is able to do this because it has access to the Master World and one or more Secondary Worlds. The above description constitutes but one exemplary way of accomplishing this feat.

Location Beacons as a Location Provider

In one embodiment, one of the location providers comprises a location beacon that beacons or transmits information to enable a computing device to actively participate in its current context. Location beacons can comprise

standalone devices that can be retrofitted onto existing infrastructures, e.g. a smoke detector or wall outlet in order for the device to have a power source.

Fig. 11 shows an exemplary beacon 1100 that is mounted on a structure 1102. Structure 1102 can be any suitable structure such as a wall in a conference room or public place, a smoke detector, an electrical socket and the like. In the described embodiment, the location beacons are small inexpensive devices that can be permanently mounted in special locations such as conference rooms, building lobbies, airport gates, public places and the like. The beacons announce the physical location in the form of an EID and/or LUID to all mobile devices that are within range, such as laptops, tablet PCs, hand held computers, mobile phones, wearable computers and the like.

In the described embodiment, the location beacon can identify the particular locations by beaconing standard information that will be understood by the mobile computing devices. In the present example, the beacons can transmit one or two location identifier pairs comprising an EID/URL pair and a LUID/URL pair. The beacon might also transmit multiple LUIDs. The EID and LUID give the present node location in the Master World and Secondary World respectively. The URLs provide a reachable location for the Master and Secondary Worlds. For example, the URL associated with the Secondary World can give a service location that the device can use to query information about the Secondary World so that it can derive its context and take advantage of resources or services that are associated with the nodes in the Secondary World.

The beacons can also transmit a digital signature that can be used by the device to ascertain that the beacon is valid and legitimate. Any suitable signature or verification method could be used. In addition, and of particular use in the

1 context-aware environment, the beacon can be programmed to transmit code
2 download pointers to devices within range. The code download pointers can
3 enable the computing device to access software code that permits them to interact
4 with their environment. Consider the following example: You walk into a
5 conference room with your cell phone computing device and immediately a
6 beacon in the conference room transmits your location in the form of an EID/URL
7 pair and a LUID/URL pair. Your device uses the information pairs to ascertain its
8 location in the Master and Secondary Worlds as described above. The beacon also
9 transmits a code download pointer that points to software code that enables you to
10 operate the video projector in the conference room using your hand-held cellular
11 phone. In this manner, the beacon serves as more than just a location beacon—it
12 permits you, through your computing device, to actively participate in your
13 surroundings.

14 The beacons can transmit the information in any suitable way, e.g. wireless
15 methods including infrared and radio frequencies. In one embodiment, Bluetooth
16 short range radio frequency communication can be used to provide a low cost, low
17 power alternative.

18 **Context Aware Enterprise Computing Policy**

19 **Overview**

20 Corporations or “enterprises” often manage large scale computer networks
21 that are used to enable members of the corporation or enterprise to be linked
22 together with each other and with corporate or enterprise resources that can be
23 used while in the enterprise computing environment. For example, a company can
24 typically provide various members with computing devices (i.e. computers,
25

laptops, handheld devices, mobile or portable computing devices and the like) that are to be used to assist the members in performing various job tasks. A continuing challenge for enterprises is to find ways to effectively and efficiently promulgate and enforce policy with respect to its computing devices. Enterprise policy can be considered as a collection of rules established by the enterprise and enforceable, relative to the enterprise's computing devices, to define various parameters of the computing environment. Considerations such as who or which devices have access to which resources, how resources are to be used, how the computing device is to be used, where the computing device is to be used and not to be used and the like, can all be impacted, in some way, by the policies that enterprise system administrators define.

In the past, policies have been defined and enforced largely on the basis of a user's or device's network identity and/or perhaps on the capability or capacity of the device. For example, consider the situation of a company that has multiple servers each of which being associated with different types of resources that are utilized by the company. One server may be associated with the company's financial resources. These resources might only be needed by individuals in the company's Finance Group. Thus, a policy can be defined by a system administrator that limits access to these servers based upon the user's identity (i.e. is the user in the Finance Group?) and/or the identity of the computer attempting to access the server (i.e. is the computer a dedicated Finance Group computer?). In this scenario, policy can be established based on the identity and/or physical location of the users and devices.

To a large degree, the traditional model for establishment and enforcement of enterprise policy is, in the future, going to have to yield to inventive new

1 models that take into account the flexible computing environment in which users
2 find themselves.

3 In accordance with the described embodiment, methods and systems are
4 provided for establishing and enforcing enterprise policy that take into account not
5 only the context of the computing device but, advantageously, changes in the
6 device's context. A device's context is ascertained and then policy associated with
7 that context is enforced. If and when the device's context changes, new or
8 additional policies can be enforced if appropriate. In one particular embodiment
9 that is used as an example through the remainder of the discussion, the device's
10 context comprises the device's particular location. It is to be understood and
11 appreciated, however, that location is used as but one example of a device context
12 and is used to assist the reader in understanding how the described principles can
13 be employed in one specific instance. Accordingly, other device contexts can be
14 used. As an example, the context of a computing device can be related to anything
15 in the device's environment or pertain to what is going on around the computing
16 device. Things such as ambient temperature, lighting conditions, and the like can
17 be considered as defining aspects of the computing device's context.

18 In the described embodiment, computing devices are provided with
19 policies, some of which pertain to enterprise computing. These policies can be
20 defined by system administrators in terms of their context dependence so that
21 when the context of a computing device changes, so too does the set of policies
22 that apply on the device. The policies can be expressed in terms of a common
23 abstract or logical representation of context or, in a specific example, location.
24 The common abstract or logical representation of context can advantageously be
25 provided through the use of the above-described techniques and systems that

utilize primary and secondary hierarchical tree structures to ascertain context and, in particular, location. It is to be appreciated and understood that any suitable way of ascertaining context or location can be utilized to enable policy to be enforced as described below.

In the illustrated example, each computing device has a policy engine, implemented in software, that receives context information or information associated with changes in context. Based on this received information, the policy engine evaluates various policies to provide a resultant set of policies that are enforced on the device, typically through the device's operating system. For example, a computing device might be located within a particular secure computing area that is ascertained as described above, through the use of a secondary world structure on the device. When in this secure computing area, the user of the computer is able to work freely on documents that are sensitive in nature in accordance with established policy. Upon leaving the secure computing area, however, the computing device determines its location (i.e. it determines that it is no longer located in the secure computing area), and the policy engine evaluates the policies to provide a new resultant set of policies that are now enforced. The new policies do not permit sensitive documents to be accessed or used when outside of the secure area. Alternately, sensitive portions of a document might be blanked out when outside of the secure area.

Exemplary Architecture

Fig. 12 shows an exemplary architecture or system generally at 1200 that is configured to implement a context-aware enterprise computing policy. The architecture can be implemented in any suitable hardware, software, firmware, or

1 combination thereof. In the illustrated example, the architecture is implemented in
2 software that comprises part of an enterprise computing device. As an aside, it is
3 to be appreciated and understood that while aspects of the described embodiment
4 are described in the context of an enterprise computing device, aspects of the
5 described embodiment could be implemented independent of an enterprise
6 computing device.

7 System 1200 comprises a context service 1202 that provides context
8 information or context change events. A suitable context service is described
9 above in connection with Fig. 6. A specific example of a context service in the
10 form of a location service is described above in connection with Fig. 7. A policy
11 engine 1204 is provided and is responsible for evaluating policies and
12 determining, based on the device's current context, a resultant set of policies that
13 apply. The resultant set of policies is then enforced on the device. Enforcement of
14 the policies can involve promulgating new settings or "state" to various
15 applications 1206 that can be executing on and off of the device.

16 The device or, more specifically, the policy engine 1204 can receive
17 policies from a number of different policy sources. Exemplary policies can
18 include device policies 1208 that pertain to the device and that come from the
19 owner of the device; network policies 1210 that pertain to one or more networks
20 within which the device can operate and that come from network administrators
21 that administer the networks; and enterprise policies 1212 that pertain to the
22 enterprise with which the device is associated and that come from an enterprise
23 administrator. All of these policies can be provided as inputs to the policy engine
24 which, as a result of a current device context, evaluates the policies and
25 determines a resultant set of policies that is then enforced on the device. The

1 policy engine 1204 can also receive, as inputs, user identity and attributes 1214
2 and device identity and attributes 1216. All of these inputs can be factored into an
3 evaluation that is performed by the policy engine 1204.

4 5 **Policy Authoring**

6 In one embodiment, advantages are achieved in the area of policy
7 authorship. Specifically, system administrators can now be given the opportunity
8 to author and define a rich, robust, and flexible set of policies that can be applied
9 in many and varying contexts. This constitutes a noteworthy departure from the
10 relatively inflexible systems in the past that enabled policy definition based only
11 upon user or device identity and/or perhaps the device's static location. In
12 accordance with the described embodiment, policy sets can be defined and then
13 enforced in a dynamic manner.

14 Fig. 13 shows a policy collection 1300 that comprises multiple different
15 sets of policies, with exemplary policy sets being shown at 1302, 1304, and 1306.
16 Policy set 1302 is associated with a first device context designated as "context
17 type 1"; policy set 1304 is associated with a second device context designated as
18 "context type 2"; and policy set 1306 is associated with a third device context in
19 the form of location designated as "location". A system administrator 1308 can
20 define various policies and associate those policies with a particular context type.
21 The defined policies then collectively provide a policy collection that can be
22 evaluated by a policy engine (e.g. policy engine 1204) so that as a device's context
23 changes, so too does the resultant set of policies that get applied on the device.
24 Additionally, various contexts for which policies are being authored need not be
25 known at the time the policy enforcement technology is being built.

As an example, consider the following: In a particular enterprise, certain computing devices are deemed as special computing devices that are not to leave the enterprise premises. Perhaps there are resources associated with the computing devices that are sensitive in nature. For example, an enterprise might be testing new proprietary software on the computing devices. A system administrator then defines a policy that establishes that if a special computing device leaves the enterprise premises, the computing device is to either lock up so that no one can use it, or purge itself of the proprietary software. In accordance with the described embodiment, the special computing devices are able to determine their location from a location service that is onboard the device. An exemplary location service is described above. If and when a computing device leaves the enterprise premises, the location service provides the location information to the device's policy engine 1204 (Fig. 12). For example, if the device moves to a new node within an applicable Secondary World, this information is used to trigger a policy re-evaluation. The policy engine then re-evaluates the policy collection 1300 to provide a new resultant set of policies that are specific to the device's new location. The new policies are now enforced through the device's operating system as described above. In the described embodiment, policy enforcement is secure because the policy engine is a trusted resource of the operating system. Enforcement in this specific example involves either locking up the device so that no one can use it, or permanently removing the proprietary software from the device.

Fig. 14 is a flow diagram that describes steps in a policy authoring method in accordance with the described embodiment. The method can be implemented in

any suitable hardware, software, firmware, or combination thereof. In the illustrated example, the method is implemented in software.

Step 1400 provides a common abstract or logical representation of context. In the illustrated example, this step is implemented through the use of the primary and secondary hierarchical structures described above. In this specific example, the primary and secondary hierarchical structures comprise the Primary and one or more Secondary Worlds. Step 1402 expresses one or more policies as a function of the common abstract or logical representation of context. In the location example, the policies are expressed as a function of the abstract representation of location, i.e. the Primary and/or one or more Secondary Worlds. This step can be implemented by a system administrator defining the policies that are to comprise a policy collection. Step 1404 then makes the expressed policies available to one or more computing devices. In the illustrated example, the computing devices comprise enterprise computing devices. The policy can be made available to the devices in any suitable way. For example, the policies can be provided to the devices via an enterprise network, wirelessly etc.

Dynamic Evaluation and Enforcement of Policies

Once the policy collection has been defined as described above, it can now be used as the basis of an adaptable, dynamic, context-sensitive policy evaluation and enforcement environment. The policy collection can have policies defined in terms of, among other variables, physical and/or logical locations. Physical locations can be provided by the Primary World, while physical and logical locations can be provided by one or more Secondary Worlds.

1 Fig. 15 is a flow diagram that describes steps in a policy evaluation and
2 enforcement method in accordance with the described embodiment. The method
3 can be implemented in any suitable hardware, software, firmware, or combination
4 thereof. In the illustrated example, the method is implemented in software that is
5 executing on one or more computing devices, such as an enterprise computing
6 device.

7 Step 1500 provides a policy collection comprising one or more sets of
8 policies. Each set of policies can have one or more individual policies. The
9 policies can be expressed in terms of a common abstract or logical representation
10 of context as described above. The policies can be stored on the enterprise
11 computing devices and can be acquired by the computing devices in any suitable
12 way. Alternately, the policies can be stored on an accessible computing device
13 such as a server or the like. Step 1502 determines the context of the computing
14 device. This step can be implemented using the techniques described above
15 insofar as the device has software that is able to process context information
16 provided to it from context providers. In a specific implementation, the context
17 information is provided in the form of location information. It is to be appreciated
18 and understood that the computing device can determine its context in more
19 indirect ways. Specifically, rather than directly determining device context
20 through the use of a context service, the computing device can determine its
21 context from other devices, networks, websites and the like. As an example,
22 consider that a computing device determines its location using a location service,
23 but desires to determine another context associated with that location. By
24 knowing the location, the computing can access a web site that maintains a list of
25 specific contexts for that particular location. Perhaps the web site tracks the

1 temperature, lighting conditions, pollen count and the like. Thus, this step can also
2 be implemented by the computing device indirectly determining its context.

3 Step 1504 evaluates the policy collection, based on the context information,
4 to provide a resultant set of policies. This step can be implemented by a policy
5 engine on the computing device. That is, this step can be implemented locally on
6 the computing device itself. Alternately, this step can be implemented remotely
7 from the computing device as by another computing device such as a server that is
8 accessible via a suitable network. In this case, it is possible for the policy engine
9 to be located remotely from the subject computing device on which the policy is to
10 be enforced. Step 1506 then enforces the resultant set of policies on the device.
11 This step can be implemented by the policy engine causing resultant settings or
12 state to be promulgated to various applications that can be executing or executable
13 on or off the device. Alternately, this step can be implemented by a remote
14 computing device, such as the server mentioned above, pushing down a resultant
15 set of policies to the computing device. Step 1508 determines whether there has
16 been a context change. This step can be implemented by the device receiving
17 context information from context providers, and then using the context
18 information to determine, based on the primary and/or secondary hierarchical tree
19 structures, whether the device has in fact experienced a context change. This step
20 can also be implemented by the device indirectly determining its context as
21 described above. If the device has experienced a context change, then the method
22 can branch back to step 1502 which determines the new context of the device and
23 repeats the steps described above. If, on the other hand, the device has not
24 experienced a context change, then the method branches back to step 1506 which
25 continues to enforce the current result set of policies.

1 This process provides a system that has policies that are dynamically
2 adaptable to new device or user contexts. Enterprise computing devices that are
3 portable can now determine, automatically, their present context and have policies
4 that are adapted to various contexts implemented in an automatic fashion. As
5 device context changes, e.g. physical or logical location, so too can the policies
6 that are enforced on the device. The policies can be expressed in terms of physical
7 and/or logical location by system administrators who now have at their disposal a
8 robust set of tools to adequately protect and administer system resources.

9 To further assist the reader in understanding the principles described above,
10 the following examples are given. It is to be appreciated and understood that the
11 following constitutes but exemplary scenarios in which the inventive principles
12 can be employed. These specific scenarios are not intended to limit application of
13 the claimed subject matter in any way.

14 Example 1

15 A particular user has an enterprise computing device with strong (e.g. 128-
16 bit) encryption capabilities. The user has to leave the United States and travel
17 abroad in foreign countries visiting various clients. Federal regulations prohibit
18 the export of a computing device with 128-bit encryption capabilities but permit
19 the export of a device with 64-bit encryption capabilities. Accordingly, when the
20 user's device determines, through the above described location service, that it has
21 left the United States, policy onboard the device causes the encryption strength to
22 be automatically downgraded.
23
24
25

1 Example 2

2 Assume that a user has a portion of the file system on their computing
3 device encrypted with a particular certificate. It is desired by the enterprise that
4 those specific files' not leave the United States. If the user's computing device
5 ascertains that its location is no longer in the United States, policy onboard the
6 computing device causes the certificate to be permanently deleted. The user must
7 then return to the U.S. and reacquire the certificate to access the portion of their
8 file system that was encrypted.

9
10 Example 3

11 A user's computing device can have policy that requires the computing
12 device to use local telephone numbers (or 1-800 numbers) when attempting to
13 establish a connection with an ISP for cost savings. When the user is out of town,
14 the policy causes the computing device to call only local numbers, associated with
15 the user's present location, to establish a connection with an ISP.

16
17 Example 4

18 The system administrator establishes a policy that when an enterprise
19 computing device is off of the corporate campus, a user login must be used and
20 that two failed attempts to log onto the device automatically locks the device
21 down. This lock down is to remain in place until the device is returned to the
22 corporate campus. Accordingly, if a user's device is stolen, it is unlikely that a
23 thief will be able to log onto the device.

1 Example 5

2 Policy is established by a system administrator that requires multiple
3 authentication methods (i.e. smartcard, password, biometric) when computing
4 from outside of the corporate campus. Accordingly, when the user removes the
5 computing device from the corporate campus, the device detects that it is no
6 longer there, and the policy engine causes the enhanced authentication to take
7 place.

8
9 Example 6

10 Policy is established by a system administrator that computer usage and the
11 location where such usage takes place must be logged for purposes of auditing or
12 perhaps billing. Accordingly, a user travels around, the policy ensures that all
13 computer usage is logged together with the location associated with the usage.

14
15 Example 7

16 The corporate enterprise has a large campus with many buildings.
17 Employees are typically required to travel between different buildings throughout
18 the day for meetings, presentations and the like. For some employees, it is critical
19 that they not miss phone calls or messages. Accordingly, the system administrator
20 authors a policy that all or selected computing devices must report their location to
21 a central server when on the corporate campus. Based on the reported location,
22 phone calls can be automatically routed directly to a conference or meeting room
23 where particular individuals are located.

1 Example 8

2 The system administrator establishes a policy that while on the corporate
3 campus, all games on enterprise computing devices are disabled. Accordingly,
4 when the computing devices determine that their location is on the corporate
5 campus, the policy engine causes all games on the device to be disabled. When
6 the computing device detects that it is no longer on the corporate campus, the
7 policy engine re-evaluates the policy collection and causes the games to be
8 enabled.

9
10 Example 9

11 A policy is established that the default printer for a computing device, when
12 the computing device is on the corporate campus, should be the physically closest
13 printer. Accordingly, a computing device determines its location and, based upon
14 this determination and the governing policy, automatically selects the closest
15 physical printer to be the default printer. As the device moves about the corporate
16 campus, the location information that is received by the policy engine causes the
17 engine to re-evaluate the policy collection to ensure that the closest physical
18 printer is selected as the default printer. This policy can be disabled when the
19 device is moved off of the corporate campus.

20
21 Example 10

22 The enterprise determines that when an employee's mobile phone and
23 mobile computer report different physical locations, there is a high degree of
24 likelihood that the employee is away from their computer. (In this example, both
25

1 computing devices are context-enabled in the sense that each can determine its
2 physical location and report that physical location to a central server).
3 Accordingly, for security purposes, the administrator authors a policy that when
4 such is the case, the login for the mobile computer is disabled. Accordingly, the
5 login is enabled only when the mobile phone and the mobile computing device
6 report the same physical location.

7 8 **Conclusion**

9 The embodiments described above provide a uniform, standardized way to
10 enhance the world of context aware computing. The embodiments provide a way
11 for individuals to uniquely experience the world around them by ascertaining their
12 location in the world in a standard way. The embodiments also provide a way for
13 service providers to uniquely position their goods and services in a manner that is
14 sensitive to and appreciates the contexts, e.g. locations, of various consumers of
15 the goods and services. Unique and useful architectures and data structures are
16 employed to facilitate the user's computing experience and provide for an
17 individual-centric experience.

18 Although the invention has been described in language specific to structural
19 features and/or methodological steps, it is to be understood that the invention
20 defined in the appended claims is not necessarily limited to the specific features or
21 steps described. Rather, the specific features and steps are disclosed as preferred
22 forms of implementing the claimed invention.